BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Setelah penyusun melakukan telaah terhadap beberapa referensi yang ada, terdapat beberapa literatur yang memiliki keterkaitan dengan perancangan yang penyusun lakukan.

Penelitian yang pertama pada Tugas Akhir yang dibuat oleh Baby Risanda dari Program Studi Teknik Elektro Sekolah Vokasi, Universitas Diponegoro pada tahun 2017 yang berjudul "Monitoring Pemakaian Daya, Arus, Tegangan, dan *Cosphi* pada KWH Meter Digital 1 Fasa Berbasis *Arduino Mega 2560* Menggunakan Aplikasi *Android*" menjelaskan tentang kWh meter digital yang dapat memonitoring pemakaian, daya, arus, tegangan, *cosphi*, serta mengkonversi pemakaian daya ke rupiah melalui aplikasi *android*. Selain itu, aplikasi android ini juga dilengkapi dengan tombol virtual on – off untuk menyala matikan kWh meter digital tersebut[1].

Penelitian yang kedua pada Tugas Akhir yang dibuat oleh Fadholi dari Program Studi Diploma III Teknik Elektro, Universitas Diponegoro pada tahun 2016 yang berjudul "Rancang Bangun ATS *(Automatic Transfer Switch)* –AMF *(Automatic Main Failure)* Pada Genset Berbasis Atmega 8 Dengan Monitoring Bahan Bakar" menjelaskan tentang sistem kerja ATS *(Automatic Transfer Switch)* –AMF *(Automatic Main Failure)* yang dikontrol menggunakan mikrokontroler berupa Atmega 8 dan memonitoring bahan bakar yang ditampilkann pada LCD 16x2[2].

Sedangkan Tugas Akhir yang akan dilakukan penyusun adalah membuat suatu aplikasi *android* yang tersambung dengan internet sebagai bentuk pemanfaatan sistem *Internet of Things* (IoT) dan dapat memonitoring tegangan, arus, status beban dengan tombol on – off untuk menyala – matikan beban dan fitur *timer* untuk *warming up* genset. Dengan begitu, maka proses monitoring pada alat akan lebih mudah karena dapat dipantau secara jarak jauh dan *real time.*

Dari perbedaan di atas terdapat beberapa perbedaan dengan penelitian yang dilakukan penyusun sebagai berikut :

1. Kedua peneliti sama – sama melakukan proses monitoring pada alat rancang bangun *(plant)* yang dikembagkan.
2. Perbedaan peneliti pertama dengan Tugas Akhir penyusun adalah peneliti pertama menampilkan pemakaian, daya, arus, tegangan, *cosphi*. Sedangkan penyusun tidak memonitoring *cosphi* seperti yang peneliti pertama lakukan.
3. Perbedaan peneliti kedua dengan Tugas Akhir penyusun adalah peneliti kedua membuat rancang bangun ATS *(Automatic Transfer Switch)* –AMF *(Automatic Main Failure)* menggunakan mikrokontroller *Atmega 8*, sedangkan penulis menggunakan *Arduino Mega 2560.* Peneliti kedua menggunakan sistem monitoring lokal yang ditampilkan melalui LCD 16x2. Sedangkan penyusun menggunakan aplikasi *Android* yang tersambung dengan internet sehingga dapat dimonitor secara jarak jauh. Serta aplikasi *android* yang dikembangkan juga dilengkapi dengan

tombol on – off virtual sebagai pengontrol nyala – mati beban dan fitur *timer* untuk *warming up* genset.

2.2. Dasar Teori

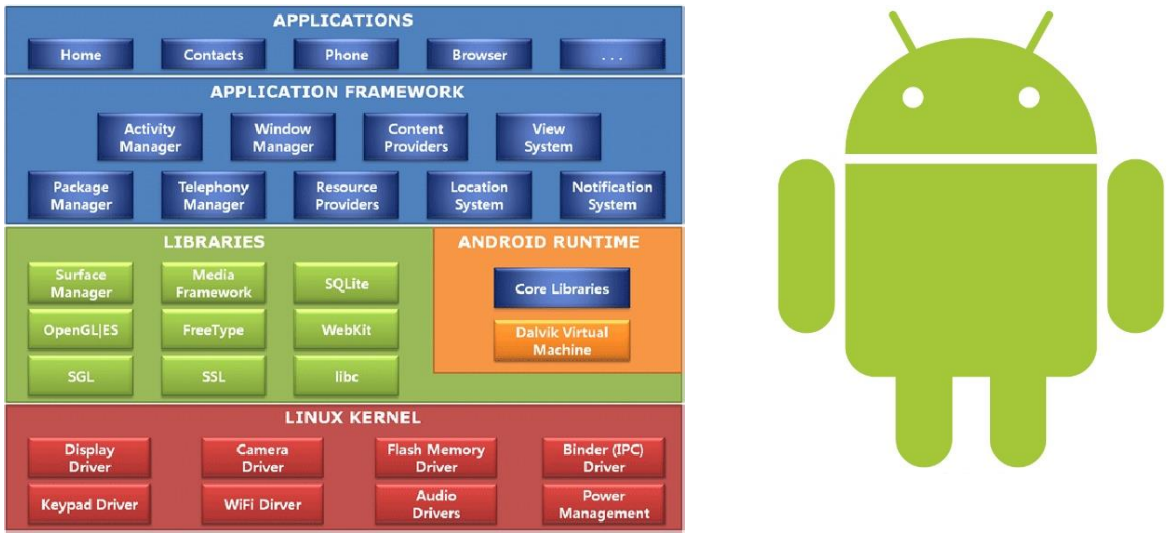
2.2.1. *Intenet of Things (IOT)*

Internet of Things, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif.

Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasikan secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT[3].

2.2.2. *Android*

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.[4]



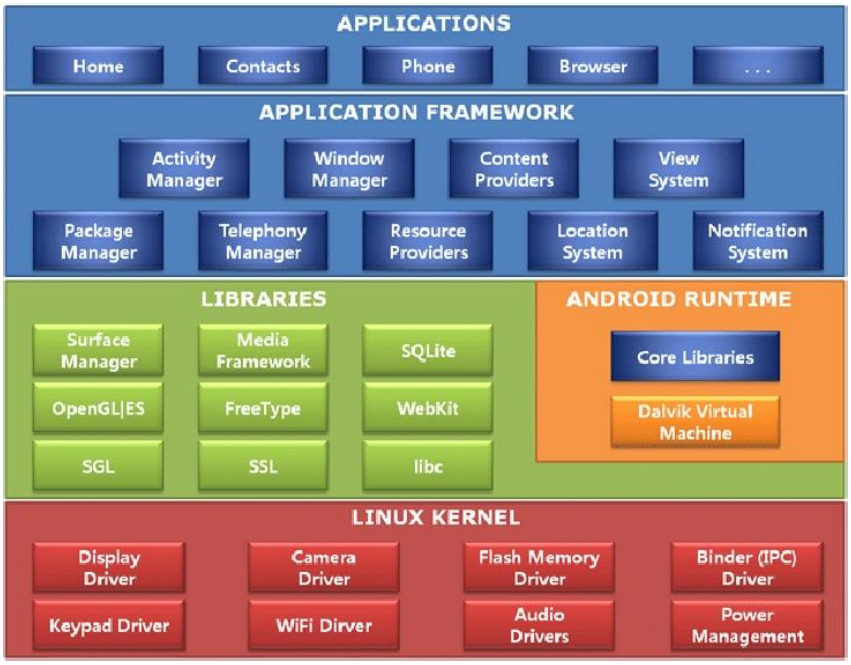
Gambar (2.1) *Android*[4]

Pada saat perilisan perdana android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Tidak hanya menjadi sistem operasi di smartphone, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android adalah karena menggunakan sistem operasi yang open source sehingga bias didistribusikan dan dipakai oleh vendor manapun dan Android itu sendiri adalah platform yang sangat lengap baik itu sistem operasinya, Aplikasi dan Tool Pengembangan, Market aplikasi Android sera dukungan yang sangat tinggi dari komunitas Open Source di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah device yang ada di dunia.

2.2.2.1. Sistem Arsitektur Sistem Operasi *Android*



Gambar 2.2 Arsitektur *Android* [4]

Sistem Operasi Android memiliki komponen utama sebagai berikut:

a. Aplikasi

Android berisi sekumpulan aplikasi utama seperti: *email client*, program *Short Message Service (SMS)*, kalender, peta, *browser*, daftar kontak, dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman Java. Pada layer inilah developer menempatkan aplikasi yang dibuat. Saat ini operasi android memiliki beberapa versi, versi terbaru adalah versi 7.0 *(Nougat).* [5]

b. Kerangka Kerja Aplikasi

Kerangka kerja aplikasi yang ditulis dengan menggunakan bahasa pemrograman *Java* merupakan peralatan yang digunakan oleh semua aplikasi, baik aplikasi bawaan dari ponsel seperti daftar kontak, dan koak SMS, maupun aplikasi yang ditulis oleh *Google* maupun pengembang *Android. Android* menawarkan para pengembang kemampuan untuk membangun aplikasi yang inovatif. Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses lokasi informasi, menjalankan *background services*, mengatur *alarm*, menambahkan peringatan ke status bar, dan masih banyak lagi. Penembang memiliki akses yang penuh ke dalam kerangka kerja API yang sama yang digunakan oleh aplikasi utama.

Pada dasarnya, kerangka kerja aplikasi memiliki beberapa komponen sebagai berikut: [5]

1) *Activity Manager*

Untuk mengatur siklus dari aplikasi dan menyediakan navigasi backstack untuk aplikasi pada proses yang berbeda.

2) *Package Manager*

Untuk melacak aplikasi yang di-install pada perangkat.

3) *Windows Manager*

Merupakan abstraksi dari bahasa pemrograman *Java* pada bagian atas dari *level service* pada *level* yang lebih redah yang disebabkan *surface manager.*

4) *Telephony Manager*

Berisi sekumpulan API yang diperlukan untuk memanggil aplikasi.

5) *Content Providers*

Memungkinkan aplikasi mengakses data dari aplikasi lain (seperti *contacts*) atau untuk membagikan data mereka sendiri.

6) *Resource Manager*

Mengakses sumber daya yang bersifat bukan kode seperti string lokal, *bitmap,* deskripsi dari *layout file* dan bagian eksternal lain dari aplikasi.

7) *Location System*

Untuk memberikan informasi detail mengenai lokasi perangkat *Android* berada.

8) *View System*

Digunakan untuk mengambil sekumpulan *button, list, grid,* dan *text box* yang digunakan antar muka pengguna.

9) *Notification System*

Mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

c. Libraries

Android memiliki sekumpulan *library C/C++* yang digunakan oleh berbagai komponen dari sistem *Android.* Kemampuan-kemampuan ini dilihat oleh para pengembang melalui kernga kerja aplikasi. Beberapa dari *library* utama dijelaskan sebagai berikut: [4]

1) *System C Library*

Merupakan implementasi turunan dari standar *system library* C *(libc)* yang diatur untuk peralatan berbasis *embedded Linux*.

2) *Media Libraries*

Disediakan oleh *PacketVideo* (salah satu anggota dari *OHA*) yang memberikan *library* untuk memutar ulang dan menyimpan format suara dan video, serta *static image file* seperti MPEG4, MP3, AAC, AMR, JPG, dan PNG.

3) *Surface Manager*

Mengatur akses kedalam subsistem tampilan dan susunan grafis layar 2D dan 3D secara mulus dari beberapa aplikasi yang menyusun permukaan gambar yang berbeda pada layar ponsel.

4) *LibWebCore*

Merupakan web browser modern yang menjadi kekuatan bagi browser *Android* dan sebuah *embeddable web view.*

5) *Sqalable Graphics Library (SGL)*

SGL mendasari mesin grafis 2D dan bekerja bersama-sama dengan lapisan pada level yang lebih tinggi dari kerangka kerja (seperti *Windows Manager* dan *Surface Manager*) untuk mengimplementasikan keseluruhan *graphics pipeline* dari *Android*.

6) *3D Libraries*

Implementasi yang didasarkan pada *OpenGL ES 1.0 APIs* dimana *library* menggunakan akselerasi perangkat keras 3D (jika tersedia), dengan rasterisasi perangkat linak 3D yang sangat optimal.

7) *Free Type Library*

Digunakan untuk menghaluskan semua tulisan bitmap dan vector.

8) *SQLite*

Merupakan relational database yang kuat dan ringan serta tersedia untuk semua aplikasi.

d. *Android Runtime*

Merupakan lokasi dimana komponen utama dari *DVM* ditempatkan. *DVM* dirancang secara khusus untuk *Android* pada saat dijalankan pada lingungan yang terbatas, dimana batrai yang terbatas, *CPU,* memori, dan penyipanan data menjadi fokus utama. *Android* memiliki sebuah *tool* yang terintegrasi yaitu "*dx*" yang mengonversi *generated byte code* dari *(.JAR)* kedalam file *(.DEX)* sehingga *byte code* menjadi lebih efisien untuk dijalankan pada prosesor yang kecil. Hal ini memungkinkan untuk memiliki beberapa jenis dari *DVM* yang berjalan pada suatu peralatan tunggal pada waktu yang

sama. *Core libraries* di tulis dalam bahasa *Java* dan berisi kumpulan *class, I/O* dan peralatan lain.

e. *Linux Kernel*

Seperti dapat dilihat pada gambar, *Linux Kernel* menyediakan driver layar, kamera, keypad, Wifi, flash memory, audio, dan IPC *(interprocess communication)* untuk mengatur aplikasi dan keamanan. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software*.

2.2.2.2. Komponen Aplikasi Android

Aplikasi *Android* ditulis dengan bahasa pemrograman *Java*. Semua file kode intermediate dan aset disatukan dalam satu paket berupa file berekstensi *.APK*, sebuah file yang dapat didistribusi. Tiap file *.APK* adalah sebuah aplikasi tunggal. Komponen aplikasi Android terdiri dari beberapa jenis, antara lain: [6]

a. *Activity*

Activity adalah istilah yang digunakan dalam pemrograman *Android* untuk mengacu pada satuan interaksi dengan pengguna melalui antarmuka grafis (*graphical user-interface, GUI*). Sebagai satuan interaksi, *Activity* adalah tampilan yang Anda lihat di layar seperti *Windows* atau kotak dialog pada pemrograman aplikasi desktop. Tiap aplikasi dapat terdiri dari nol atau lebih *Activity*. Selain sebagai satuan eksekusi, *Activity* selalu memiliki paling tidak satu buah *Thread,* yakni *Tread* utama yang digunakan untuk memperbarui tampilan *user-interface (UI Thread).*

b. Internet

Internet adalah istilah yang digunakan dalam pemrograman *Android* untuk mengacu pada mekanisme berbagi pesan pemberitahuan atau bertukar data *Activity* atau untuk menjalankan aplikasi lain.

c. *Service*

Service adalah komponen aplikasi yang berjalan di belakang layar tanpa *user-interface* untuk menyediakan layanan tertentu seperti mengecek RSS feed secara kontinu atau memainkan musik. Service tetap berjalan meski *Activity* yang mengendalikannya telah berhenti. *Media player* adalah sebuah contoh aplikasi yang menggunakan *Service*.

d. *Content Provider*

Content provider membuat suatu aplikasi dapat berbagi sejumlah data tertentu kepada aplikasi lain. Jika membutuhkan data nama-nama kontak, aplikasi tinggal meminta data tersebut.

e. *Broadcast Receiver*

Broadcast receiver adalah komponen yang memantau, menerima, dan berinteraksi terhadap pesan yang disebarkan, baik oleh sistem maupun aplikasi lain. Misalnya, ketika baterai lemah, *Android* akan mengirim pesan "baterai lemah" kepada semua *broadcast receiver* yang ingin diberitahu pesan ini. Untuk menggunakan *broadcast receiver*, pada dasarnya, hanya perlu membuat turunan tipe broadcast receiver, melengkapi metode onReceive(), dan mendaftarkannya di AndroidManifest.xml atau dengan metode Context.registerReceive(). Instance broadcast receiver hanya valid

selama pemanggilan metode onReceive() sehingga Anda tidak boleh menyimpan referensi ke instance ini. Satu hal yang dapat Anda lakukan ketika onReceive() dipanggil adalah mewakilkannya ke komponen lain, misalnya dengan memanggil metode startActivity() atau startService() milik Context.

2.2.2.3. Versi Android

Android telah mengalami sejumlah pembaruan sejak pertama kali dirilis.

Gambar 2.3 Versi *Android* [3]

Tabel 2.1 menunjukan beberapa jenis *Android* dan nama kodenya. Penamaan kode menggunakan nama makanan dan huruf depannya urut sesuai abjad. [4]

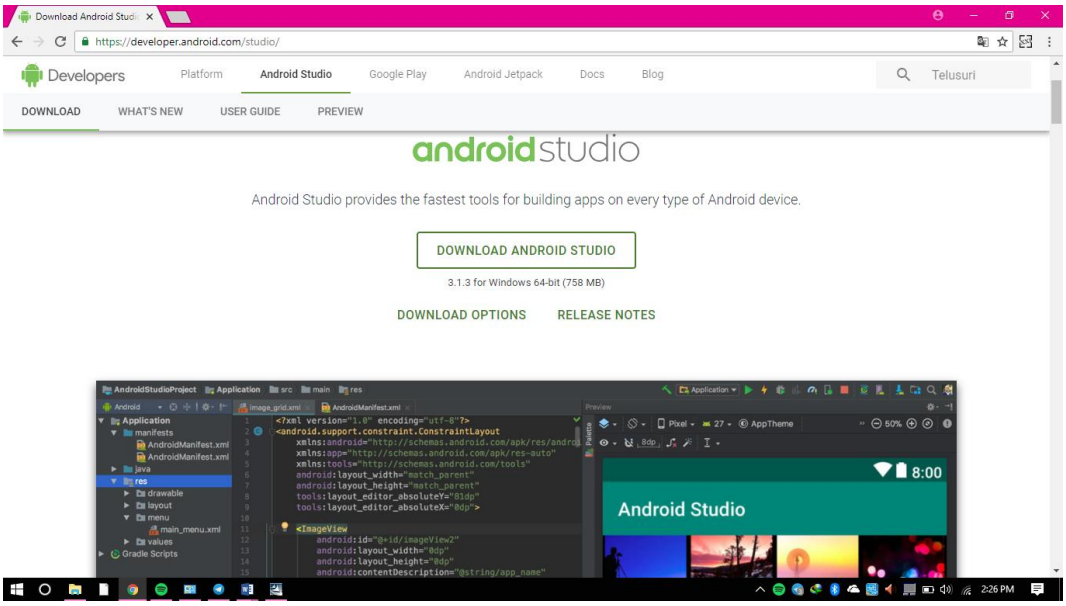
Tabel 2.1 Versi *Android* [6]

Versi	Nama	Tanggal Rilis	Level API
1.0	(Tanpa Nama)	23 september 2008	1
1.1	(Tanpa Nama)	9 Februari 2009	2
1.5	Cupcake	30 April 2009	3
1.6	Donut	15 September 2009	4
2.0	Éclair	20 Oktober 2009	5
2.0.1		3 Desember 2009	6
2.1		12 Januari 2010	7
2.2 – 2.2.3	Froyo	20 Mei 2010	8
2.3 – 2.3.2	Gingerbread	6 Desember 2010	9
2.3.3 – 2.3.7		9 Febuari 2011	10
3.0	Honeycomb	22 Februari 2011	11
3.1		10 Mei 2011	12
3.2		15 Juli 2011	13
4.0 – 4.0.2	Ice Cream Sandwich	19 Oktober 2011	14
4.0.3 – 4.0.4		16 Desember 2011	15
4.1	Jelly Bean	9 Juli 2012	16
4.2		13 November 2012	17
4.3		24 Juli 2013	18
4.4	KitKat	31 Oktober 2013	19

5.0	Lollipop	25 Juni 2014	21
5.1		12 November 2014	22
6.0	Marshmallow	17 Agustus 2015	23
7.0	Nougat	23 Agustus 2016	24
7.1		4 Oktober 2016	25
8.0	Oreo	21 Agustus 2017	26

2.2.3. Peranti Pengembangan Aplikasi *Android*

Pertumbuhan pasar perangkat berbasis *Android* tentu saja mendorong pertumbuhan pengembangan aplikasi berbasis *Adroid*. Bagi sisi pengembang, peranti yang memudahkan pembuatan aplikasi tentu saja diharapkan. Beruntung sekali, situs *Android Developers* (developer.android.com), lihat Gambar 2.4 menyediakan *Android Studio* yang mempermudah siapapun untuk membuat aplikasi *Android*. [6]

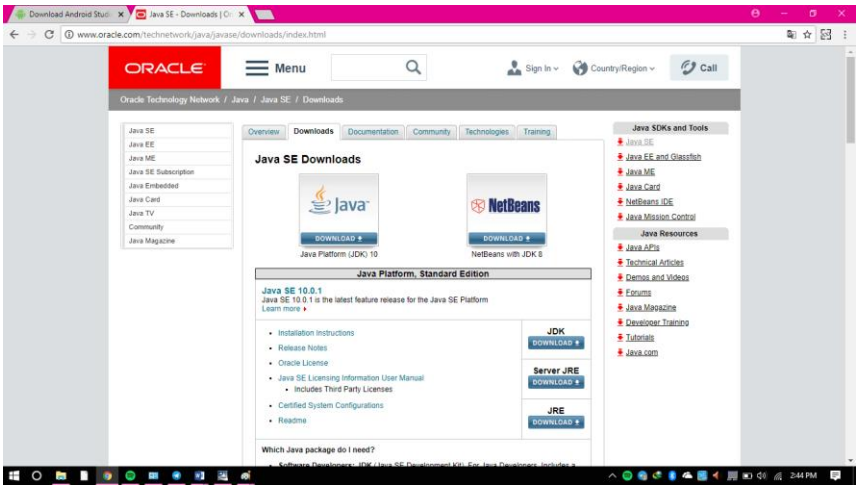


Gambar 2.4 Situs *Android Development* [7]

Peranti yang diperlukan untuk pengembangan aplikasi *Android* mencakup *Java Development Kit (JDK)* dan *Android Studio*.

2.2.3.1. *Java Development Kit (JDK)*

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *Java* menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Environtment*. Perangkat ini mutlak diperlukan untuk membuat aplikasi *Android*, mengingat aplikasi *Android* itu berbasis *Java.* Sebagaimana diketahui, *Java* adalah suatu bahasa pemrograman yang biasa digunakan untuk membuat aplikasi. Namun, perlu diketahui, tidak semuapustaka dalam Java digunakan di *Android*. Sebagai contoh, *Android* tidak menggunakan *Swing*. [7]



Gambar 2.5 Situs *Oracle* yang menyediakan fasilitas Pengunduhan *JDK* [7]

2.2.3.2. *Android Studio*

Android Studio adalah *IDE (Integrated Development Environment)* resmi untuk pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran *Android Studio* ini diumumkan oleh *Google* pada 16 Mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, *Android Studio* mengantikan *Eclipse* sebagai *IDE* resmi untuk mengembangkan aplikasi *Android*.[8]

Gambar 2.6 *Android Studio* [8]

Android studio sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* disertai dengan *ADT plugin (Android Development Tools)*. *Android Studio* memiliki fitur:

a. Projek berbasis pada *Gradle Build.*
b. *Refactory* dan pembenahan bug yang cepat.
c. *Tools* baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
d. Mendukung *Proguard And App-signing* untuk keamanan.

e. Memiliki *GUI* aplikasi android lebih mudah.

Didukung oleh *Google Cloud*

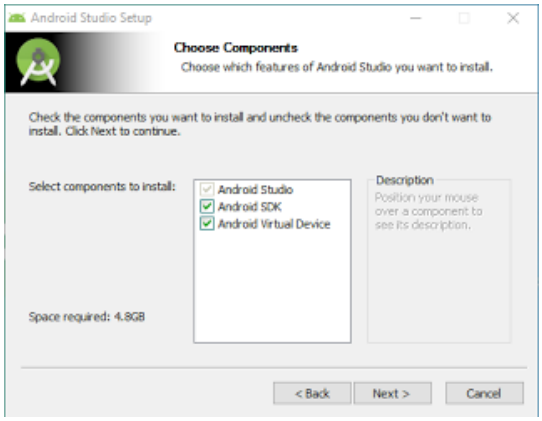
Cara menginstall *Android Studio :*

1) *Download Android Studio* pada website resminya di https://developer.android.com/studio/index.html.

2) Kemudian *double click* installer *Android Studio*, tunggu sampai *initializing* selesai dan akan muncul *system administration permission*. Klik *Ok* dan tampilah jendela *setup* seperti gambar dibawah ini. Selanjutnya klik *Next*.

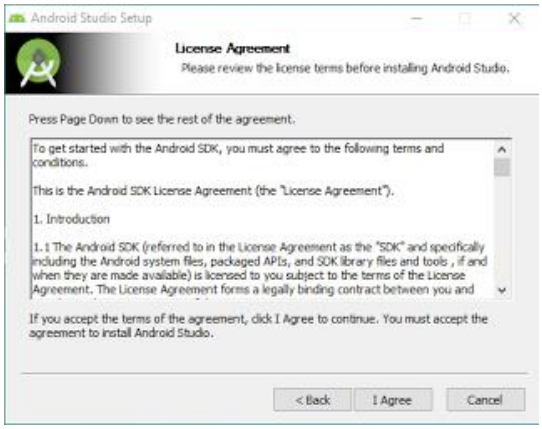
Gambar 2.7 *Android Studio Set – up*

3) Lalu pilih komponen yang akan di *install*. Disana otomatis semua langsung di *check list*, klik *Next.*



Gambar 2.8 *Choose Components*

4) Pada *License Agreement* klik "*I Agree".*

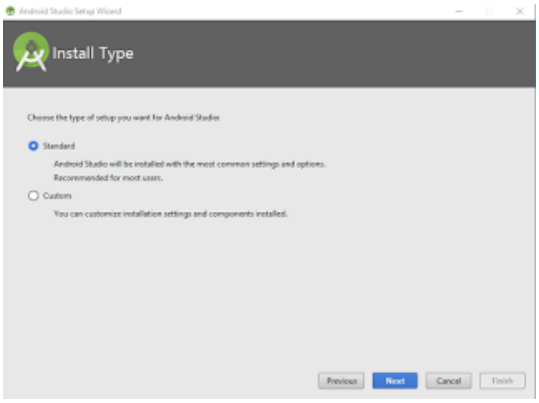


Gambar 2.9 *License Agreement*

5) Selanjutnya pemilihan lokasi penyimpanan file-file *Android Studio*, lalu klik *Next*. Kemudian klik *Install.*
6) Tunggu installasi hingga selesai. Jika installasi sudah selesai, klik *Next.*
7) Untuk melengkapi instalasi silahkan klik *Finish.*
8) Kemudian akan muncul tampilan *Android Studio.* Tunggu beberapa saat hingga selesai.

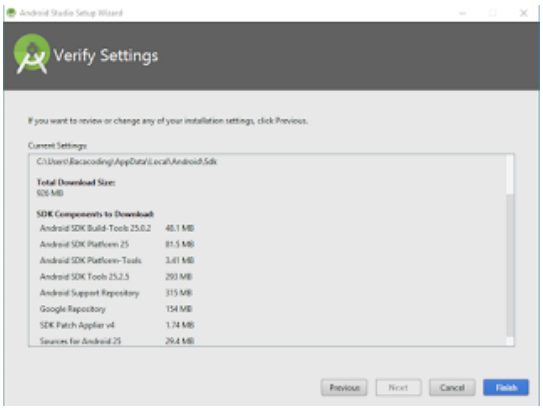
9) Lalu munul halaman pembuka *Android Studio*, klik *Next.*

10) Kemudian pilih tipe installasi *Standart* atau *Custom*, disarankan untuk memilih *Standart*. Lalu klik *Next.*



Gambar 2.10 *Install Type*

11) Lalu muncul beberapa file yang akan didownload oleh *Android Studio*, klik *Finish.*



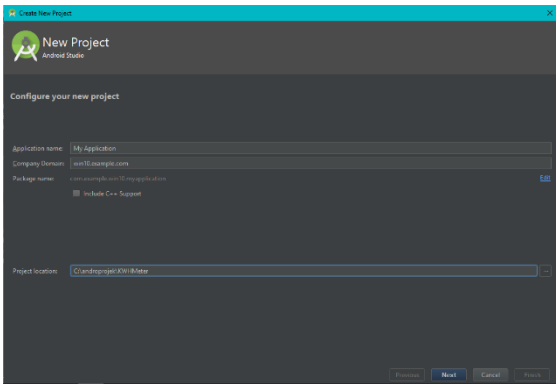
Gambar 2.11 *Verify Settings*

12) Tunggu proses *Downloading* dan *Extracting* selesai. Lama nya proses tergantung koneksi dan laptop.

13) Jika suah selesai maka, akan tampil jendela *Android Studio*. Dan installasi sesungguhnya pun selesai. Selanjutnya klik *Start a new Android Studio Project* untuk membuat *project* baru.

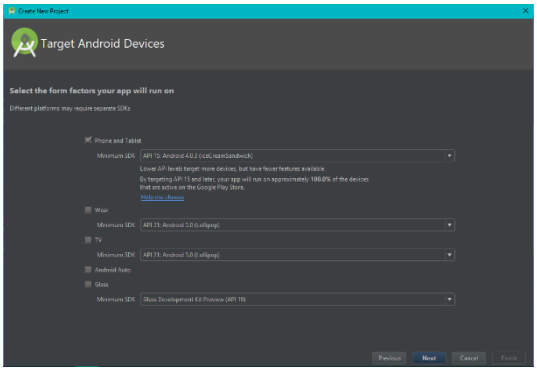
Gambar 2.12 *Welcome to Android Studio*

14) Kemudian konfigurasi *project* yang dibutuhkan, jika sudah klik *Next.*



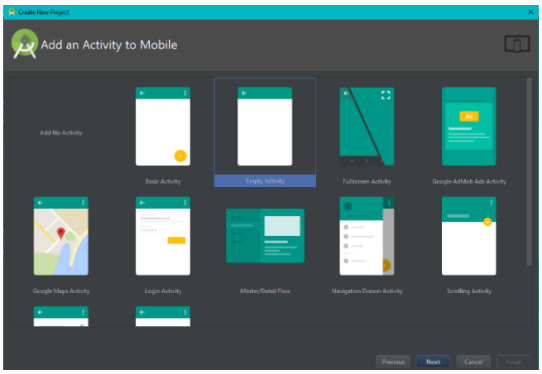
Gambar 2.13 *New Project*

15) Selanjutnya pilih target perangkar *Android* yang diinginkan. Kemudian pilih minimum SDK nya, yaitu *Ice Cream Sandwich*



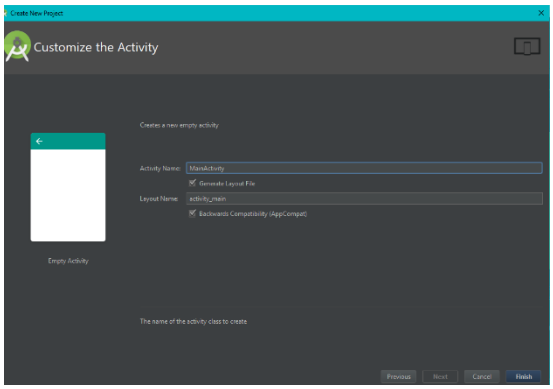
Gambar 2.14 *Target Android Device*

16) Pilih *empty activity* pada *add on activity to mobile.* Jika sudah klik *Next.*



Gambar 2.15 *Add an Activity to Mobile*

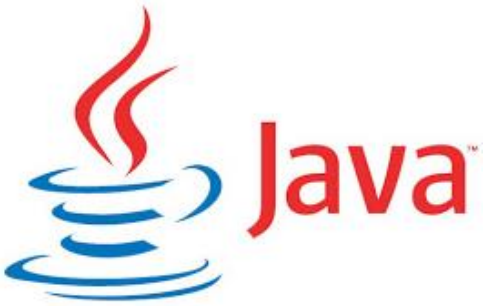
17) Kemudian beri nama aktifitasnya, klik *Finish*. *Android Studio* siap digunakan.



Gambar 2.16 *Customize the Activity*

2.2.4 *Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer termasuk ponsel. Bahasa ini awalnya dibuat oleh James Gosling saan masih bergabung dengan *Sun Microsystems* (yang saat ini bagian dari *Oracle*) dan dirilis tahun 1995. Kata *Java* yang dipakai sebagai nama bahasa pemrograman ini konon dipilih Gosling karena kesukaannya terhadap kopi yang biasa Ia minum. Menurut berbagai sumber, kopi itu berasal dari Pulau Jawa (Java adalah kata bahasa Inggris untuk Jawa). [9]



Gambar 2.17 *Java* [10]

Bahasa *Java* mengadopsi sintaksis-sintaksis pada *C* dan *C++*, namun dengan sintakis model objek yang lebis sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi-aplikasi berbasis *Java* umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan di berbagai *Java Virtual Machine (JVM). Java* merupakan bahasa pemrograman bersifat umum/nonspesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya, aplikasi *Java* mampu berjalan di beberapa platform sistem operasi yang berbeda. *Java* dikenal pula dengan sloganya: "Tulis sekali, jalankan di mana pun". Saat ini *Java* merupakan bahasa pemrograman terpopuler dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.2.5.2. Karakteristik *Java*

Sebagai sebuah bahasa pemrograman, *Java* dapat membuat seluruh bentuk aplikasi seperti desktop, web, ataupun yang lainnya. *Java* merupaka bahasa pemrograman yang berorientasi objek dan dapat dijalankan di berbagai platform sistem operasi. Perkembangan *Java* tidak hanya terfokus pada suatu sistem operasi dan bersifat open source. Berikut karakteristik Java yang dikemukakan secara resmi oleh Sun: [10]

a. Sederhana (*Simple*)

   Bahasa pemrograman *Java* menggunakan sintaksis yang mirip dengan bahasa pendahulunya, yaitu C++. Akan tetapi, ada beberapa perubahan yang dilakukan

seperti hilangnya penggunaan *pointer* yang rumit dan *multi-inheritance*. Selain itu, *Java* menggunakan *automatic memory allocation* dan *garbage collection.*

b. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek sehingga memungkinkan sebuah program dibuat secara modular dan bisa dipergunakan kembali.

c. Terinterpretasi (*Interpreted*)

Java dijalankan dengan interpreter, yaitu *Java Virtual Machine (JVM).* Dengan adanya *JVM, source code Java* telah dikompilasi menjadi *bytecode* dapat dijalankan di platform yang berbeda-beda.

d. Terdistribusi (*Distributed*)

Java membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintgrasi pada *Java*.

e. Reliabel

Java mempunyai reliabilitas tinggi. Compiler Java berkemampuan untuk mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. *Java* mempunyai *runtime exception* handling untuk membantu mengatasi error pada pemrograman.

f. Netral arsitekturnya

Program Java merupakan *platform* independen. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan *JVM.*

g. Aman (*Secure*)

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, *Java* memiliki beberapa mekanisme keamanan agar aplikasi tidak digunakan untuk merusak sistem computer yang menjalankan aplikasi tersebut.

h. Mampu ditingkatkan performansinya

Performnsi *Java* sering dikatakan kurang tinggi. Namun, performansi pada *Java* dapat ditingkatkan degan kompilasi Java lain seperti buatan *Inprise, Microsoft,* ataupun *Symantec* yang menggunakan *Just-In-Time Compiler (JIT).*

i. Portabel

Source code dan program *Java* dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus di kompilasi ulang.

j. Multithreaded

Java berkemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara bersamaan.

k. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan dinamis. Perubahan pada suatu class dengan penambahan property ataupun method dapat dilakukan tanpa mengganggu program pengguna class tersebut.

2.2.5 *XML (Extensible Markup Language*)

XML adalah markup language yang dikembangkan oleh *World Wide Web Consortium (W3C),* dengan tujuan utamanya adalah untuk mengatasi sejumlah

keterbatasan yang terdapat pada *Hyper Text Markup Language (HTML). XML* dan *HTML* merupakan subset dari *Structured Generalized Markup Language* (SGML). Secara aktual *XML* lebih mirip *SGML* dibandingkan dengan *HTML*, karena *HTML* hanya digunakan untuk mendiskripsikan web pages. Tetapi *XML* adalah language yang digunakan untuk mendiskripsikan dan memanipulasi struktur dokumen, serta menawarkan beberapa mekanisme untuk memanipulasi informasi yang bebas *platform*. Sebagai contoh, *XML* digunakan oleh *StarOffice* dan *AbiWord* untuk salah satu format penyimpanan dokumen dan *XML* digunakan untuk menyimpan obyek persisten dalam dokumen perkantoran. *XML* berkonsentrasi pada struktur informasi, tetapi tidak berkonsentrasi untuk menampilkan dokumen informasi. [11]

XML dirancang khusus unntuk penyampaian informasi melalui *World Wide Web (WWW),* sama seperti *HTML* yang telah menjadi bahasa standar untuk membuat halaman web sejak awal kehadiran web. *XML* adalah salah satu format/ekstensi file yang berbasis text, yang memiliki ekstensi berakhiran *(.xml).* Penggunaan *XML* untuk pemrograman web interaktif sangat cocok sekali, selain mudah dimengerti struktur elemennya karena menggunakan tag sesuai keinginan kita sendiri, begitu juga dengan *Script*-nya (menggunakan *JavaScript, Jscript atau VBScript).*

2.2.5.1. Keuntungan *XML*

Beberapa keuntungan yang didapat dari file *XML*, diantaranya:

a. Ekstensibilitas, dapat ditukar atau digabung dengan dokumen *XML* lain.

b. Pemrograman yang lebih aik maka dibuat suatu *software* pengolah *XML*.
c. Memisahkan data dan presentasi yang akan direpresentasikan dalam *XML* dan *XSLT (Extensible Stylesheet Language Transformation).*
d. Pencarian data yang cepat karena *XML* merupakan data dalam format yang terstruktur.
e. *Plain text* dan *platform independent*.
f. Untuk pertukaran data.

Dokumen *XML* dapat digunakan untuk berbagai macam tujuan, seperti:

a. Sebagai pengimpan data (*database*) yang mudah dibaca oleh pengguna karena disimpan dengan bentuk text.
b. Standar transfer data, dapat digunakan untuk pengiriman data transaksi antar perusahaan atau mengirim data dari *DBMS (Database Management* System) yang berbeda (misalnya dari *Oracle* ke *SQL Server).*

2.2.6. *Arduino Mega 2560*

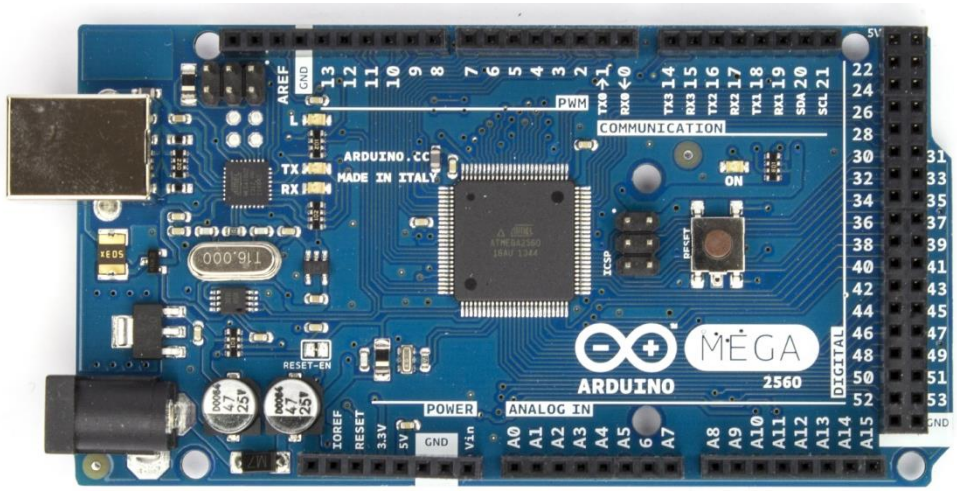
Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan ATmel.

Mikrokontroler itu sendiri adalah *chip* atau *Integrated Circuit (IC)* yang bisa diprogram menggunakan komputer. Tujuan ditanamkannya program pada mikrokontroler adalah supaya rangkaian elektronik dapat membaca *input*, kemudian memproses *input* tersebut sehingga menghasilkan *output* yang sesuai

dengan keinginan. Jadi mikrokontroler berfungsi sebagai otak yang mengatur *input*, proses, dan *output* sebuah rangkaian elektronik.

Arduino Mega 2560 adalah papan mikrokontroler berbasiskan *Atmega 2560* yang memiliki 54 pin digital *input/output*, dimana 15 pin diantaranya digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, 4 pin sebagai UART (port *serial hardware)*, sebuah osilator kristal 16 MHz, koneksi USB, *jack* power, header ISCP, dan tombol *reset*[1].

Gambar 2.18 adalah tampilan board *Arduino Mega 2560*. Dan tabel 2.2 menunjukkan spesifikasi dari *Arduino Mega 2560*



Gambar 2.18 *Arduino Mega 2560 dan Spesifikasinya [1]*

Tabel 2.1 Spesifikasi dari *Arduino Mega 2560*[1]

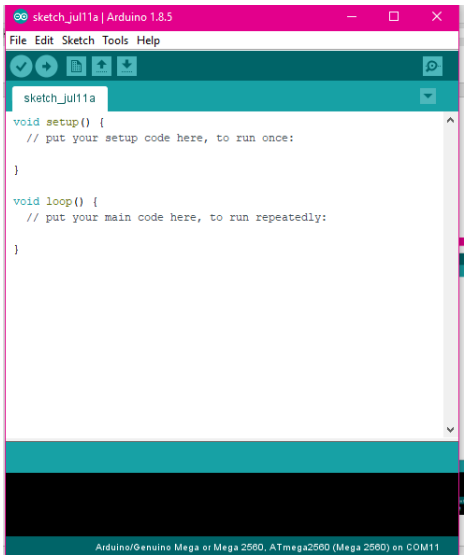
Mikrokontroler	ATmega2560
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V

Input Voltage (limit)	6-20V
Pin Digital I/O	54 (yang 15 pin digunakan sebagai *output* PWM)
Pins *Input* Analog	16
Arus DC per pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	256 KB (8 KB digunakan untuk *bootloader*)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

(Sumber: www.arduino.cc, diakses tanggal 16 Juni 2016)

2.2.7. Aplikasi Program *Arduino IDE*

Ketika kita membuka program *Arduino IDE (Integrated Developed Environment)*, Akan muncul tampulan kerja seperti gambar 2.19 dibawah ini. Pada tampilan tersebut merupakan Program *Arduino IDE* pada Windows 10. Pada dasarnya *IDE (Integrated Development Environment)* akan sama pada Operasi Sistem apapun yang digunakan.



Gambar 2.19 *Tampilan program IDE*

(Integrated Development Enviroment)

IDE terpisah dari toolbar, The code ada ditengah dan The Serial Output ada dibawah terdiri dari enam tombol diantaranya :

a. Verify / compile

Digunakan untuk mengecek atau memeriksa apakah kode sudah benar sebelum di kirim kepapan Arduino.

b. Upload

Mengirim lembar kerja kedalam papan Arduino.

c. New

Berfungsi untuk membuat tampilan lembar kerja atau sketch baru untuk memasukan kode.

d. Open

Menampilkan list lembar kerja yang telah di simpan.

e. Save

Menyimpan lembar kerja atau sketch.

f. Serial Monitor

Menampilkan hasil data-data yang telah dikirim dari Arduino.

Untuk memulai Serial Monitor, tekan tombol Serial Monitor dan untuk menghentikan tekan tombol Stop. Pada Linux, Arduino akan me-reset sendiri ketika meng-klik tombol Serial Monitor. Untuk mengoprasikan atau menggabungkan Arduino pada PC (Personal Computer), kita dapat menggunakan program – program seperti Processing, Flash, MaxMSP, Visual Basic, dan lain – lain.

2.2.7.1. Menu *Softwere Arduino*

Kita dapat melihat beberapa menu untuk mengakses software arduino diantaranya adalah:

1. **Menu Help**

   Pada menu ini dapat membantu kita menemukan informasi lebih lagi tentang IDE *(Integrated Development Enviroment).*

   1. *Getting Started*
   2. *Environment*
   3. *Reference*
   4. *Galileo Help*
   5. *Getting Started*
   6. *Troubleshooting*

7. *Edison Help*
8. *Getting Started*
9. *Troubleshooting*
10. *Find in reference*
11. *Frewuently Ask Questions*
12. *Visit Arduino.cc*
13. *About Arduino*

b. **Menu *Tools***

a. *Auto Format*
b. *Archive sketch*
c. *Fix Encoding & Reload*
d. *Serial Monitor*
e. *Serial Plotter*
f. *Wifi101 Firmware Update*
g. *Board*
h. *Processor*
i. *Port*
j. *Get Board Info*
k. *Programmer*
l. *Burn Bootlaoder*

c. Menu *Sketch*

1. *Verify/Compile*
2. *Upload*

3. *Upload Using Programmer*
4. *Export Compile Binary*
5. *Show Sketch Folder*
6. *Include Library*
7. *Add File*

d. **Menu *Edit***

1. *Undo*
2. *Redo*
3. *Cut*
4. *Copy*
5. *Copy for Forum*
6. *Copy at HTML*
7. *Paste*
8. *Select All*
9. *Go to Line*
10. *Comment / Uncommernt*
11. *Increase Indent*
12. *Decreas Indent*
13. *Increase Font Size*
14. *Decrease Font Size*
15. *Find*
16. *Find Next*
17. *Find Previous*

e. **Menu *File***

1. *New*
2. *Open*
3. *Open Recent*
4. *Sketchbook*
5. *Examples*
6. *Close*
7. *Save*
8. *Save As*
9. *Page Setup*
10. *Print*
11. *Preference*
12. *Quit*

2.2.7.2. Pemrograman Bahasa C *Arduino*

C adalah bahasa yang standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi. Standar bahasa C yang asli adalah standar dari UNIX. Sistem operasi, kompiler C dan seluruh program aplikasi UNIX yang esensial ditulis dalam bahasa C. Patokan dari standar UNIX ini diambilkan dari buku yang ditulis oleh Brian Kerninghan dan Dennis Ritchie berjudul "The C Programming Language", diterbitkan oleh Prentice-Hall tahun 1978. Deskripsi C dari Kerninghan dan Ritchie ini kemudian dikenal secara umum sebagai "K&R C".

Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut menjadi standar, ANSI (American National Standards Institute) membentuk suatu komite (ANSI committee X3J11) pada tahun 1983 yang kemudian menetapkan standar ANSI untuk bahasa C. Standar ANSI ini didasarkan kepada standar UNIX yang diperluas. Standar ANSI menetapkan sebanyak 32 buah kata-kata kunci (keywords) standar. Versi-versi bahasa C yang menyediakan paling tidak 32 kata-kata kunci ini dengan sintaks yang sesuai dengan yang ditentukan oleh standar, maka dapat dikatakan mengikuti standar ANSI. Buku ajar ini didasarkan pada bahasa C dari standar ANSI.[3]

2.2.7.3. Program C *Arduino*

a. Struktur

Setiap program *Arduino* (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada.

a. *void setup*( ) { }

   Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

b. *void loop*( ) { }

   Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan. [3]

2. Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

a. //(komentar satu baris)

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

b. **/* */**(komentar banyak baris).

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

c. **{ }**(kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

d. **;**(titk koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

3. *Variabel*

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

1. **int** (integer)

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak

mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.

2. **long** (long)
   Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari - 2,147,483,648 dan 2,147,483,647.

3. **Boolean** (Boolean)

   Variabel sederhana yang digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

4. **Float** (float)
   Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.
5. **Char** (character)

   Menyimpan 1 karakter menggunakan kode ASCII (misalnya ‘A’ = 65). Hanya memiliki 1 byte (8 Bit) dari RAM.

4. Operator Matematika

   Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

   1. =

      Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: x = 10 * 2, x sekarang sama dengan 20).

2. **%**

   Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya : 12% 10, ini akan menghasilkan angka 2).

3. **+**

   Penjumlahan

4. **–**

   Pengurangan

5. *****

   Perkalian

6. /

   Pembagian

5. Operator Pembanding

   Operator Pembanding digunakan untuk membandingkan nilai logika.

   1. ==

      Sama dengan (misalnya: 12 == 10 adalah FALSE (salah) atau 12 == 12 adalah TRUE (benar)).

   2. !=

      Tidak sama dengan (misalnya: 12 != 10 adalah TRUE (benar) atau 12 != 12 adalah FALSE (salah))

   3. **<**

      Lebih kecil dari (misalnya: 12 < 10 adalah FALSE (salah) atau 12 < 12 adalah FALSE (salah) atau 12 < 14 adalah TRUE (benar))

4. >

   Lebih besar dari (misalnya: 12 > 10 adalah TRUE (benar) atau 12 > 12 adalah FALSE (salah) atau 12 > 14 adalah FALSE (salah))

f. Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan.

1. if..else, dengan format seperti berikut ini:

   if (kondisi) { }

   else if (kondisi) { }

   else { }

   Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya FALSE maka kode pada *else* yang akan dijalankan.

2. for, dengan format seperti berikut ini:

   for (int i = 0; i < #pengulangan; i++) { }

   Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan *i*++ atau ke bawah dengan *i*–.

g. Digital

1. pinMode(pin, mode)

   Digunakan untuk menetapkan mode dari suatu pin, *pin* adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah *INPUT* atau *OUTPUT*.

2. digitalWrite(pin, value)

   Ketika sebuah pin ditetapkan sebagai *OUTPUT*, pin tersebut dapat dijadikan *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

h. digitalRead(pin)

Ketika sebuah pin ditetapkan sebagai INPUT maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

i. Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

1. analogWrite(pin, value)

   Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (*on*)atau mati (*off*) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. *Value* (nilai) pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0V) dan 255 (100%

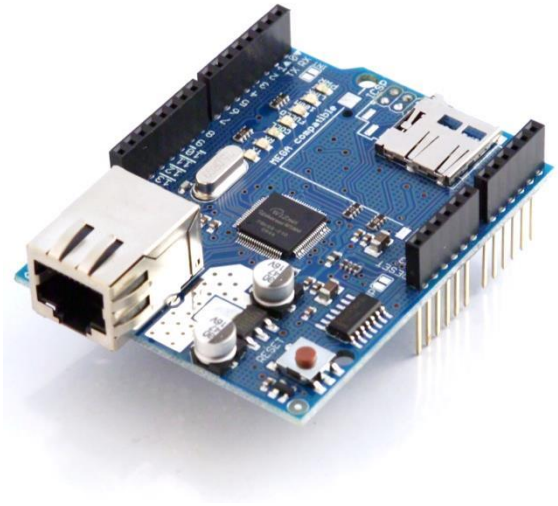
duty cycle ~ 5V).

2. analogRead(pin)

   Ketika pin analog ditetapkan sebagai *INPUT* anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).[3]

2.2.8. *Ethernet Shield*

Ethernet Shield menambah kemampuan arduino board agar terhubung ke jaringan komputer. Perangkat *Ethernet Shield* ditunjukkan pada gambar 2.20.



.

Gambar 2.20 *Ethernet Shield*[12]

Ethernet shield berbasiskan *chip* ethernet *Wiznet W5100*. Ethernet *library* digunakan dalam menulis program agar arduino board dapat terhubung ke jaringan dengan menggunakan *ethernet shield*. Pada *ethernet shield* terdapat sebuah slot *micro-SD,* yang dapat digunakan untuk menyimpan file yang dapat diakses melalui

jaringan. Onboard *micro-SD* card reader diakses dengan menggunakan *SDlibrary*. Arduino board berkomunikasi dengan *W5100* dan *SD card* mengunakan bus *SPI (Serial Peripheral Interface).* Komunikasi ini diatur oleh library *SPI.h* dan *Ethernet.h*.

Bus SPI menggunakan pin digital 11, 12 dan 13 pada *Arduino Uno* dan pin 50, 51, dan 52 pada Mega. Pin digital 10 digunakan untuk memilih *W5100* dan pin digital 4 digunakan untuk memilih *SD card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk input/output umum ketika kita menggunakan ethernet shield. Karena *W5100* dan *SD card* berbagi bus *SPI*, hanya salah satu yang dapat aktif pada satu waktu.

Jika kita menggunakan kedua perangkat dalam program kita, hal ini akan diatasi oleh *library* yang sesuai. Jika kita tidak menggunakan salah satu perangkat dalam program kita, kiranya kita perlu secara eksplisit mendeselect-nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai output dan menuliskan logika tinggi padanya, sedangkan untuk *W5100* yang digunakan adalah pin 10.

Untuk menghubungkan *ethernet shield* dengan jaringan, dibutuhkan beberapa pengaturan dasar. Yaitu *ethernet shield* harus diberi alamat MAC *(Media Access Control)* dan alamat IP *(Internet Protocol)*. Sebuah alamat MAC adalah sebuah identifikasi unik secara global untuk perangkat tertentu. Alamat IP yang valid tergantung pada konfigurasi jaringan. Hal ini dimungkinkan untuk menggunakan *DHCP (Dynamic Host Configuration Procotol)* untuk secara dinamis menentukan sebuah IP. Selain itu juga diperlukan *gateway* jaringan dan *subnet*[12].

2.2.9. Modem

Modem Berasal dari singkatan *Modulator Demodulator*. *Modulator* merupakan bagian yang mengubah sinyal informasi ke dalam sinyal pembawa (*carrier*) dan siap untuk dikirimkan, sedangkan *Demodulator* adalah bagian yang memisahkan sinyal informasi (yang berisi data atau pesan) dari sinyal pembawa yang diterima sehingga informasi tersebut dapat diterima dengan baik. Modem merupakan penggabungan kedua-duanya, artinya modem adalah alat komunikasi dua arah. Setiap perangkat komunikasi jarak jauh dua-arah umumnya menggunakan bagian yang disebut "modem", seperti VSAT, *Microwave Radio*, dan lain sebagainya, namun umumnya istilah modem lebih dikenal sebagai Perangkat keras yang sering digunakan untuk komunikasi pada komputer.

Data dari komputer yang berbentuk sinyal digital diberikan kepada modem untuk diubah menjadi sinyal analog, ketika modem menerima data dari luar berupa sinyal analog, modem mengubahnya kembali ke sinyal digital supaya dapat diproses lebih lanjut oleh komputer. Sinyal analog tersebut dapat dikirimkan melalui beberapa media telekomunikasi seperti telepon dan radio.

Setibanya di modem tujuan, sinyal analog tersebut diubah menjadi sinyal digital kembali dan dikirimkan kepada komputer. Terdapat dua jenis modem secara fisiknya, yaitu modem eksternal dan modem internal. [12]

Gambar 2.21 *Modem*[12]

2.2.10. Router TP – Link TL – MR3420

TP-Link TL-MR3220 merupakan 3G / 3.75G Wireless Router N yang memungkinkan pengguna untuk berbagi koneksi mobile broadband 3G / 3.75G dengan keluarga dan teman-teman kapan saja dan dimana saja. Dengan standar Wireless Router N dan Antena 5 dBi Omni-directional, menjadikan TP-Link TL-MR3220 wireless router yang dapat diandalkan untuk berbagi pengalaman berselancar internet yang menyenangkan. Modem ini dapat disuplai dengan tegangan 12VDC.

Gambar 2.22 Router TP – Link TL – MR3420

(Sumber :

https://www.cleverboxes.com/media/catalog/product/cache/1/image/9df78eab335

25d08d6e5fb8d27136e95/O/R/OR101200000164009.jpg

diakses pada tanggal 11 Juli 2018)

2.2.11. Teleduino

Teleduino adalah Web Server yang digunakan untuk menghubungkan arduino dengan aplikasi lain menggunakan media internet. Web server ini dapat mengontrol arduino dengan menggunakan API Key. Setiap perangkat yang ingin dihubungkan ke web server teleduino harus merequest api key dan api key itu akan di masukkan kedalam mikrokontroller. Sehingga satu api key hanya khusus Teleduino adalah Web Server yang digunakan untuk menghubungkan arduino dengan aplikasi lain menggunakan media internet. Web server ini dapat mengontrol arduino dengan menggunakan API Key. Setiap perangkat yang ingin dihubungkan ke web server teleduino harus merequest api key dan api key itu akan di masukkan kedalam mikrokontroller. Sehingga satu api key hanya khusus ditujukan untuk 1 mikrokontroller yang ditanamkan juga api key tersebut di dalamnya.

Teleduino juga harus di definisikan terlebih dahulu pin mana saja pada mikrokontroller yang akan digunakan sehingga dapat disesuaikan dan dapat dihubungkan ke aplikasi yang ingin kita buat. Dari sinilah aplikasi dan mikrokontroller terhubung dengan syarat keduanya mendapatkan akses internet.[3]

Untuk masuk ke teleduino tersebut cukup mudah dengan membuka situs teleduino resmi yaitu https://www.teleduino.org/. Dalam situs tersebut ada beberapa pilhan menu yaitu :

1. Introduction berisikan menu penjelasan mengenai web server teleduino.
2. Request Key merupakan menu yang disediakan untuk merequest sebuah API Key yang akan kita gunakan sehingga web server teleduino akan terhubung dengan Mikrokontroller yang kita gunakan.
3. Tools memiliki 2 menu pilihan yaitu Arduino Sketch Key digunakan untuk menghasilkan kode yang dapat disalin dengan cara mengisi kotak yang ada dengan API Key yang sudah di request sebelumnya. Manage Presets Alat ini digunakan untuk mengatur nilai preset pada perangkat Teleduino Anda. Nilai preset ini ditetapkan saat perangkat di-booting.
4. Documentations Berisikan beberapa menu yaitu Installation, API dan Tutorials.
5. Downloads
6. Terms and Conditions
7. Sponsor
8. Contac

Dalam web server teleduino tersebut kita bisa langsung memberikan sebuah pengaturan yang akan diberikan pada Mikrokontroller arduino. Pengaturan tersebut berupa pendefinisian yang akan dilakukan oleh arduino dan pin berapa yang akan digunakan. Pengaturan tersebut bisa berupa set digital output, Get all Inputs dan lain lain. Adapun daftar definisi definisi

yang dapat kita lakukan di teleduino melalui API Key yaitu :

System :

1. Reset
2. getVersion
3. setStatusLedPin
4. setStatusLed
5. getFreeMemory
6. ping
7. getUptime
8. loadPresets

I/O :

1. definePinMode
2. setDigitalOutput
3. setPwmOutput
4. getDigitalInput
5. getAnalogInput
6. getAllInput
7. setDigitalOutputs

Shift Registers :

1. defineShiftRegister
2. setShiftRegister
3. mergeShiftRegister

4. getShiftRegister

Serial :

1. defineSerial
2. getSerial
3. setSerial
4. flushSerial

Servo :

1. defineServo
2. setServo

EEPROM :

1. Reset Eeprom
2. setEeprom
3. getEeprom

Wire (TWI/I2C) :

a. defineWire
b. setWire
c. getWire

2.2.12. *Charger*

Charger sering juga disebut *converter* adalah suatu rangkaian peralatan listrik yang digunakan untuk mengubah arus listrik bolak balik (*Alternating Current*, disingkat AC) menjadi arus listrik searah (*Direct Current*, disingkat DC), yang berfungsi untuk pasokan DC *power* baik ke peralatan-peralatan yang menggunakan sumber DC maupun untuk mengisi baterai agar kapasitasnya tetap

terjaga penuh sehingga keandalan unit pembangkit tetap terjamin. Dalam hal ini baterai harus selalu tersambung ke *rectifier.*

Kapasitas *rectifier* harus disesuaikan dengan kapasitas baterai yang terpasang, setidaknya kapasitas arusnya harus mencukupi untuk pengisian baterai sesuai jenisnya yaitu untuk baterai alkali adalah 0,2 C (0,2 x kapasitas) sedangkan untuk baterai asam adalah 0,1C (0,1 x kapasitas) ditambah beban statis (tetap) pada unit pembangkit. Sebagai contoh jika suatu unit pembangkit dengan baterai jenis asam kapasitas terpasangnya adalah 200 Ah dan arus statisnya adalah 10 *Ampere*, maka minimum kapasitas arus *rectifier* adalah[13]:

$I_{rectifier}$ = (0,1 x 200Ah)+ 10 A

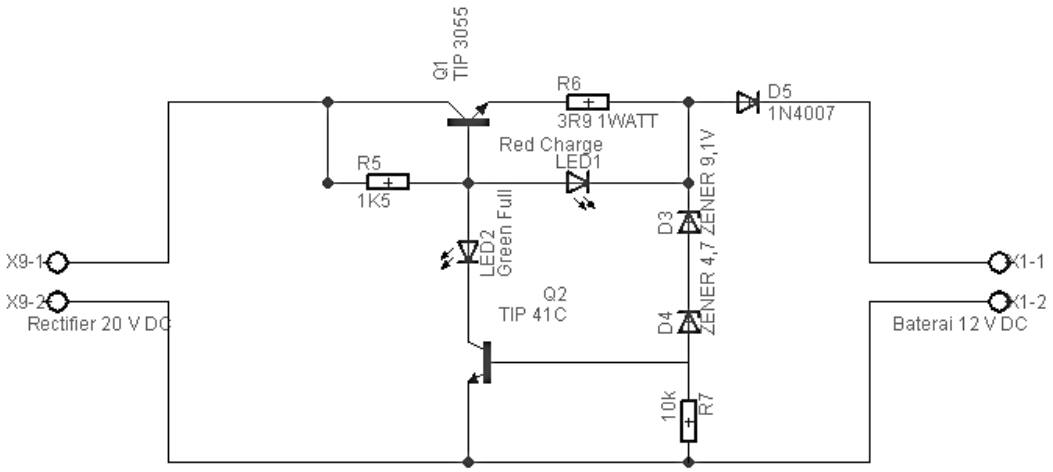
= 20 A + 10 A

= 30 Ampere

Jadi, kapasitas *rectifier* minimum yang harus disiapkan adalah sebesar 30 *Ampere*[13].

2.2.12.1. Cara Kerja Charger

Pada gambar 2.23 ditunjukkan rangkaian charger untuk baterai kering 12 V DC tipe *Lead Acid*.



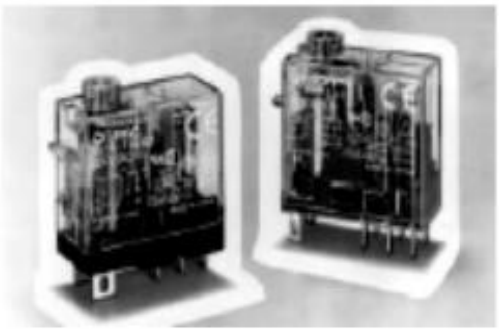
Gambar 2.23 Rangkaian *Charger*

Rangkaian charger pada gambar 2.41 menggunakan 2 transistor, 2 LED, 2 dioda zener, diode, dan 3 resistor. Cara kerja charger di atas adalah saat breakdown zener tercapai yaitu hampir 13,8V maka dioda zener akan terbuka dan mengalirkan arus ke Basis Q2, maka Q2 mulai *On* atau mengalirkan arus dari kolektor ke emitor. Led hijau mulai menyala sedangkan led merah meredup. Saat Q2 mengalirkan arus dari kolektor ke emitor secara penuh maka led hijau menyala terang, led merah padam dan Q1 akan *off*. Saat baterai penuh, transistor Q1 TIP3055 akan off atau tidak mengalirkan arus dari kolektor ke emitor. Saat voltase breakdown zener tercapai maka zener akan terbuka dan memicu transistor untuk On, arus akan dibuang ke *ground*. Karena arus dibuang ke ground maka arus menuju baterai akan menjadi sangat kecil dan tetap menjaga voltase pada batas yang ditentukan saja. Setelah baterai terpakai, maka voltase baterai menurun, zener

kembali menutup, dan proses charging berjalan kembali sampai cut off tercapai[14].

2.2.13. *Relay*

Dalam dunia elektronika, *relay* dikenal sebagai komponen yang dapat mengimplementasikan logika *switching.* Sebelum tahun 70an, *relay* merupakan "otak" dari rangkaian pengendali. Baru setelah itu muncul *PLC* yang mulai menggantikan posisi *relay*. *Relay* yang paling sederhana ialah *relay* elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. [15]



Gambar 2.24 *Relay* yang tersedia di pasaran [15]

Secara sederhana *relay* elektromekanis ini didefinisikan sebagai berikut:

a. Alat yang menggunakan gaya elektromagnetik untuk menutup atau membuka kontak saklar.
b. Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

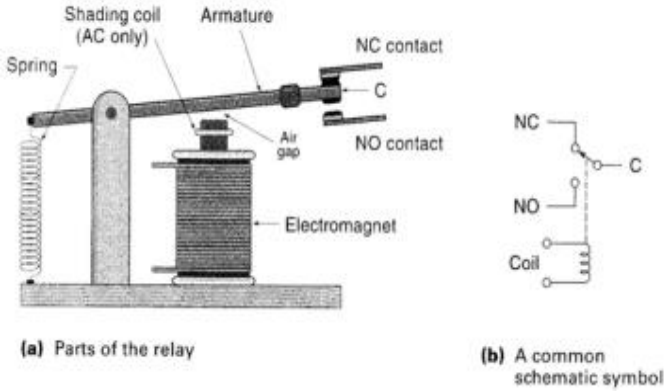
Secara umum, *relay* digunakan untuk memenuhi fungsi – fungsi berikut:

a. *Remote control*: dapat menyalakan atau mematikan alat dari jarak jauh

b. Penguatan daya: menguatkan arus atau tegangan. Contohnya adalah *starting relay* pada mesin mobil.

c. Pengatur logika kontrol suatu sistem

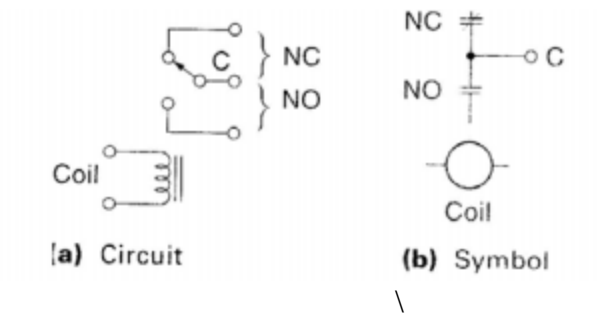
2.2.13.1. Prinsip Kerja dan Simbol



Gambar 2.26 Skema relay elektromekanik [15]

Relay terdiri dari *coil* dan *contact*. Perhatikan gambar (2-22), *coil* adalah gulungan kawat yang mendapat arus listrik, sedang *contact* adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contac*t ada 2 jenis: *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*).

Secara sederhana berikut ini prinsip kerja dari *relay*: ketika *coil* mendapat energi listrik akan timbul gaya elektromagnet yang akan menarik armature yang berpegas, dan *contact* akan menutup.



Gambar 2.27 Rangkaian dan simbol logika *relay* [15]

Selain berfungsi sebagai komponen elektronik, *relay* juga mempunyai fungsi sebagai pengendali sistem. Sehingga *relay* mempunyai 2 macam simbol yang digunakan pada:

a. Rangkaian listrik (*hardware).*
b. Program *(software).*

2.2.13.2. Jenis – Jenis *Relay*

Seperti saklar, *relay* juga dibedakan berdasar *pole* dan *throw* yang dimilikinya. Berikut definisi *pole* dan *throw*:

Pole: banyaknya contact yang dimiliki oleh relay.

Throw: banyaknya kondisi *(state)* yang mungkin dimiliki *contact.*

Berikut ini penggolongan *relay* berdasar jumlah *pole* dan *throw*:

a. *SPST (Single Pole Single Throw) Relay* golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*.
b. *DPST (Double Pole Single Throw) Relay* golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar

sedangkan 2 Terminal lainnya untuk *Coil. Relay DPST* dapat dijadikan 2 Saklar yang dikendalikan oleh 1 *Coil*.

c. *SPDT (Single Pole Double Throw) Relay* golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil.*

d. *DPDT (Double Pole Double Throw) Relay* golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang *Relay SPDT* yang dikendalikan oleh 1 *(single) Coil.* Sedangkan 2 Terminal lainnya untuk *Coil.*